

Pixel Detector Size and Shape

David Christian
Fermilab

February 5, 2001

Context of Discussion

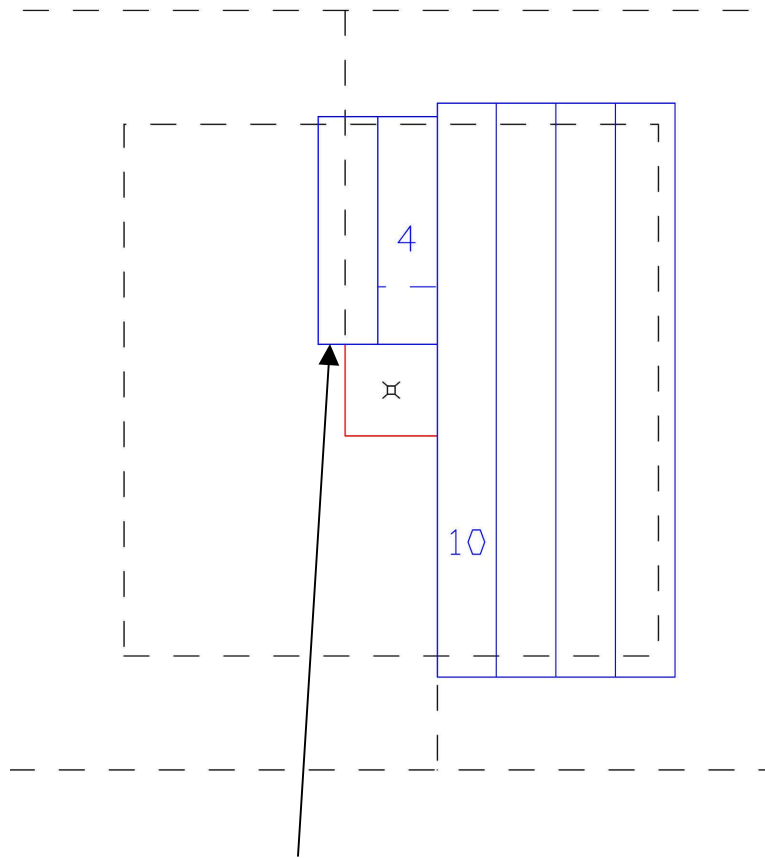
- FPIX2 Design
 - “Core” established by PreFPIX2I, 2TB
 - Programming interface & DAC’s established by PreFPIX2TB
 - Final step is periphery design
 - SEU considerations
 - Data output (point to point LVDS)
 - Data format (BCO, Col, Row-ADC, Row-ADC?)
 - Degree of serialization (# of lines)
 - Array size
- Geometry specification for next round of BTeV simulations
 - Hope is to have geometry files ready for use in ~3-6 weeks.

Starting point...

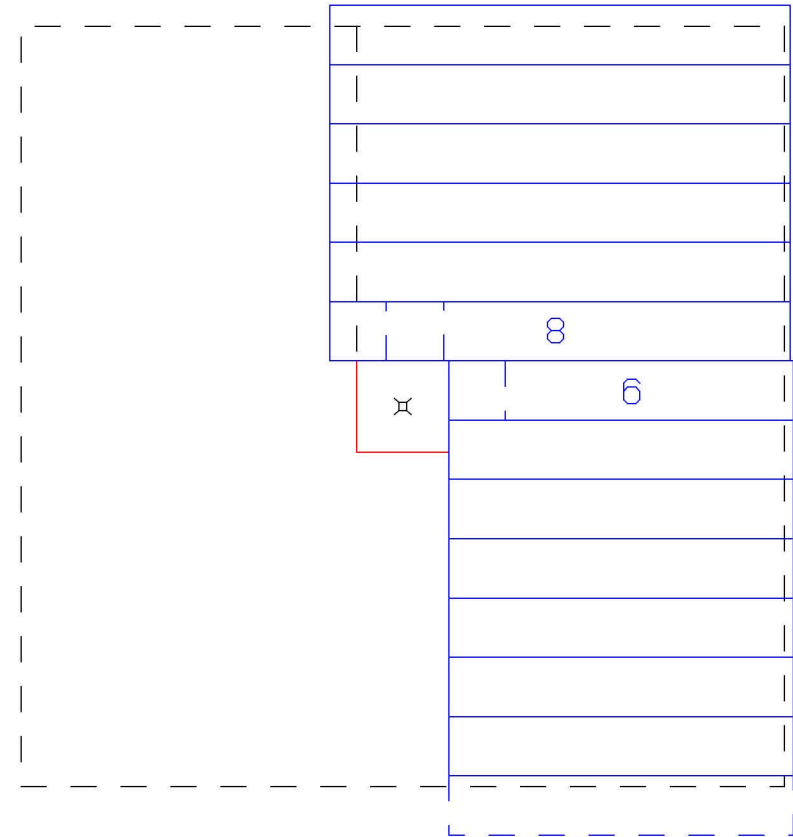
- 12 mm x 12 mm beam hole.
- Vertical separation of half planes (tracks from beam region never cross central vertical plane).
- Baseline trigger uses precision non bend view pixels only for inner triplets.
- Penny's simulations (need a presentation) indicate that precision non bend view coverage of 70% of bend view coverage is sufficient (not really relevant for today's discussion).

(160 row x 18 col) size doesn't tile nicely

Non-bend "view"



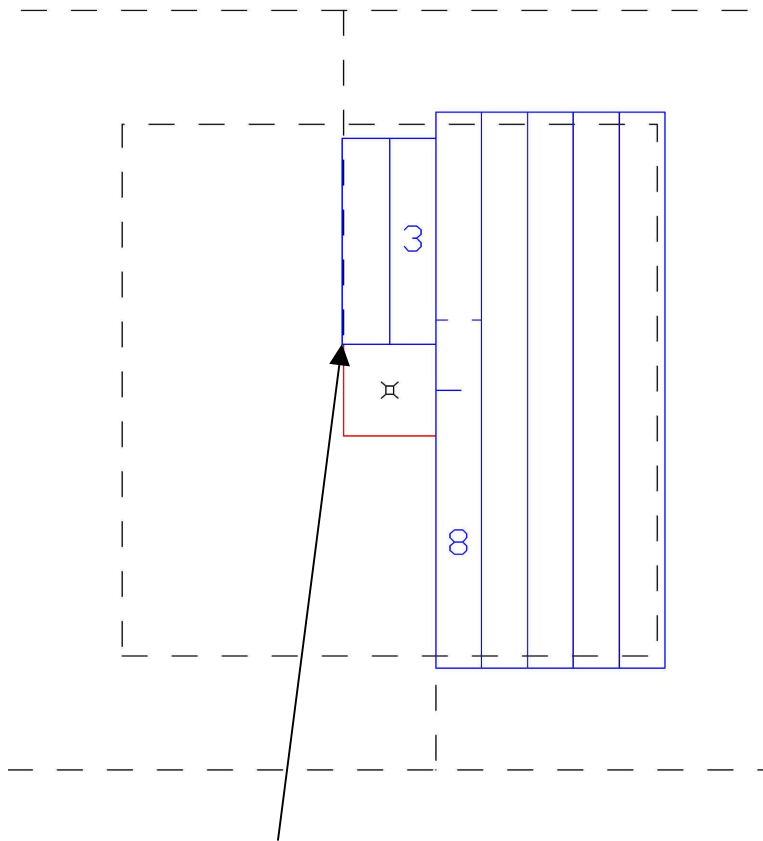
Bend "view"



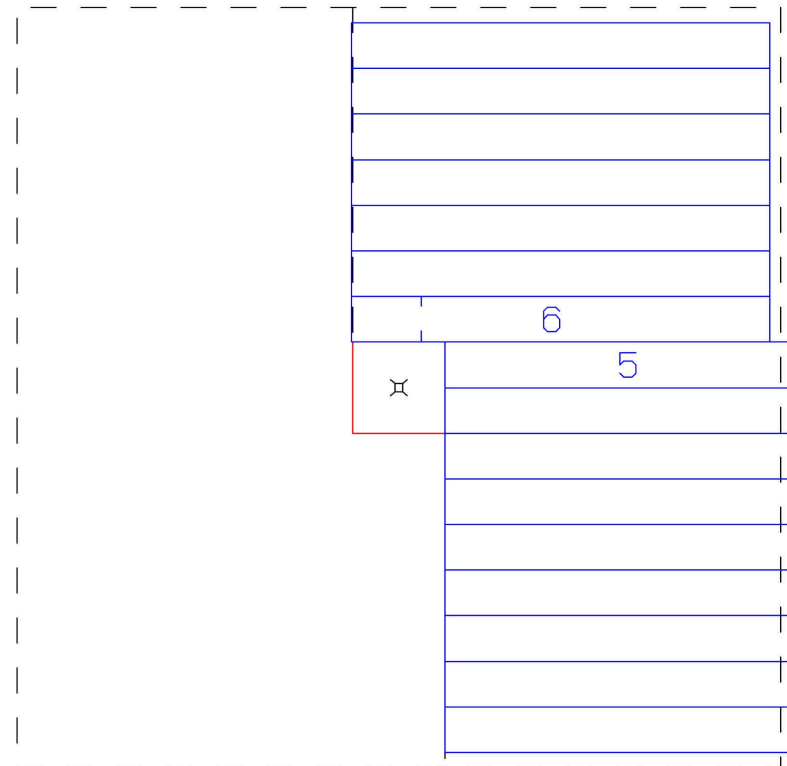
3.2 mm overlap, assuming 8 row shingle overlap (7.6 mm coverage)

(128 x n) works nicely (128x22 shown)

Non-bend “view”



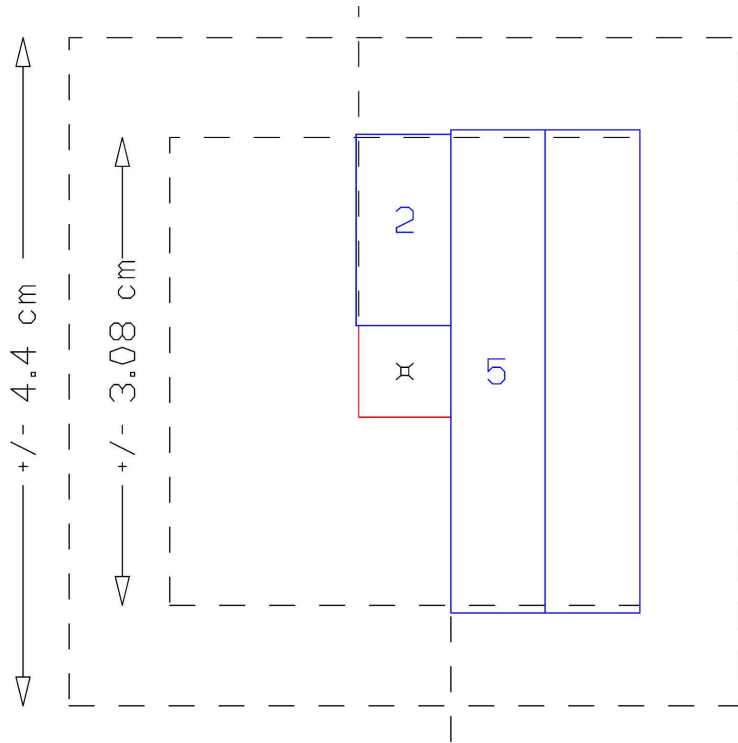
Bend “view”



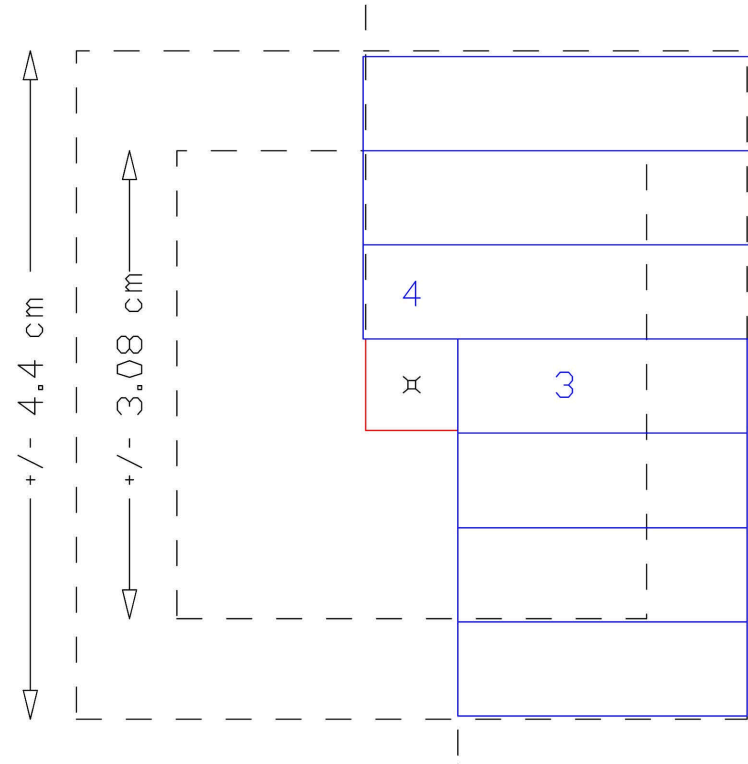
Very small overlap, or none for 8 row shingle overlap (6 mm coverage)

(256 x n) works well too.

Non-bend “view”



Bend “view”



Concentrate on 128x & 256x

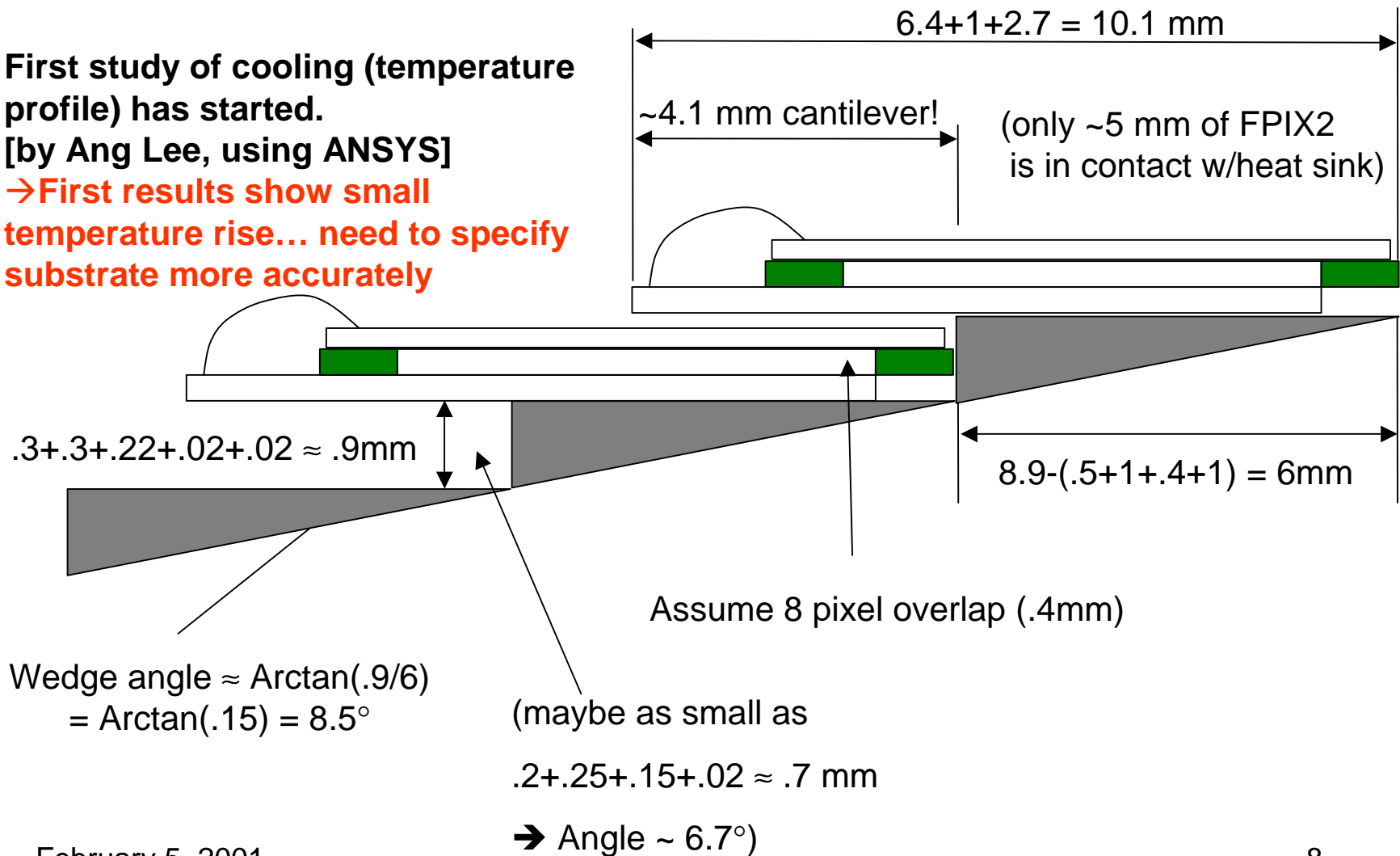
- Shingling (wedge angle required)
- Constraints on HDI?
- FPIX2 considerations
- Cooling

Wedge required for sensor w/128 rows

First study of cooling (temperature profile) has started.

[by Ang Lee, using ANSYS]

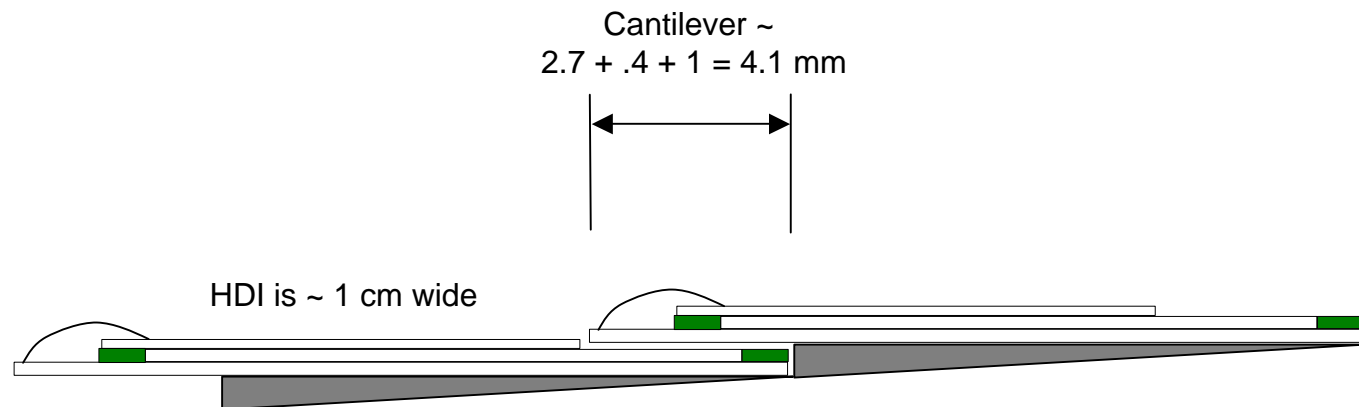
→ First results show small temperature rise... need to specify substrate more accurately



256 x n: wedge angle is small.

Cooling is either the same as
(128 x n), or easier.

(first results → problem is the same)



$$\begin{aligned}\text{Wedge angle} &\approx \text{Arctan}(.7/12.4) \\ &= \text{Arctan}(.06) = 3.4^\circ\end{aligned}$$

FPIX2 array size = 12.4mm x 12.8mm;
Total chip size \approx 12.8mm x 16.5mm

FPIX2 considerations: (160x18) vs. (128x22)

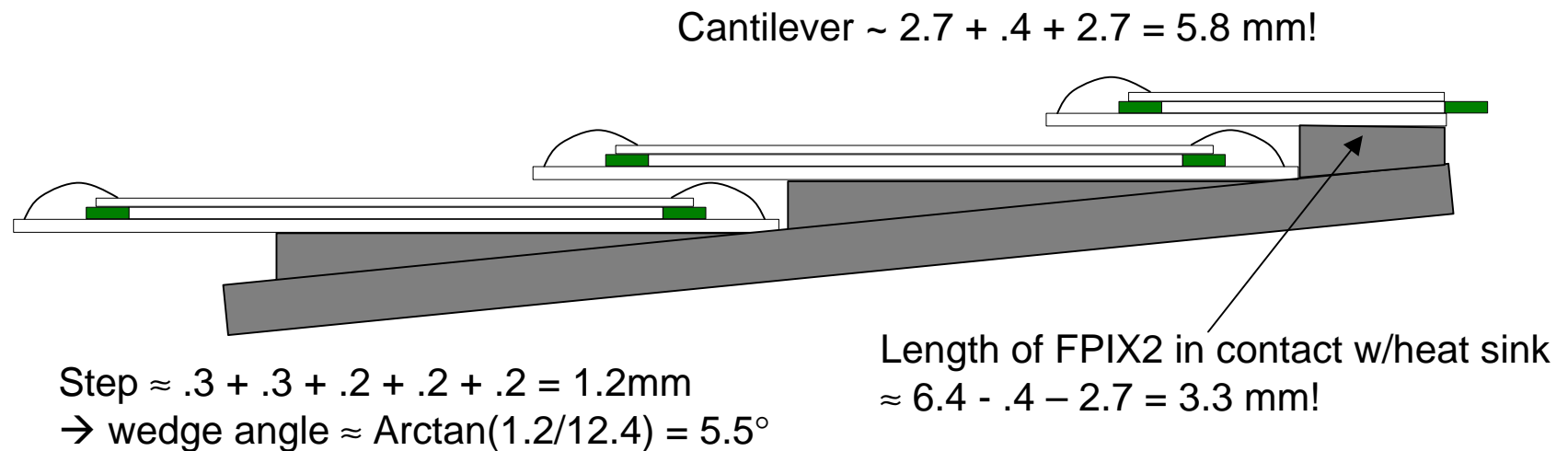
- Simulations → (160 x 18) can (just) be read out fast enough to maintain high efficiency.
 - May have to extend periphery more than current 2.7 mm (or add columns) to fit serializers & LVDS drivers. (Adding columns increases bandwidth required.)
 - Starting to simulate (128x22):
 - ~The same area.
 - Can use faster readout clock.
 - Added space on periphery to accommodate serializers & LVDS drivers.
- (128x22) looks **very nice** from readout chip
Point of view.

FPIX2 Considerations: (256xn)

- Can't simply extend the column without using Metal 6 for power distribution.
- Metal 6 is available in IBM 0.25; not TSMC.
- Can't extend the column without redesigning token-passing or using a much slower readout clock.
- Only easy option is to use 128 rows & layout "2 chips in 1" – back to back (as intended before concept of shingle was introduced).

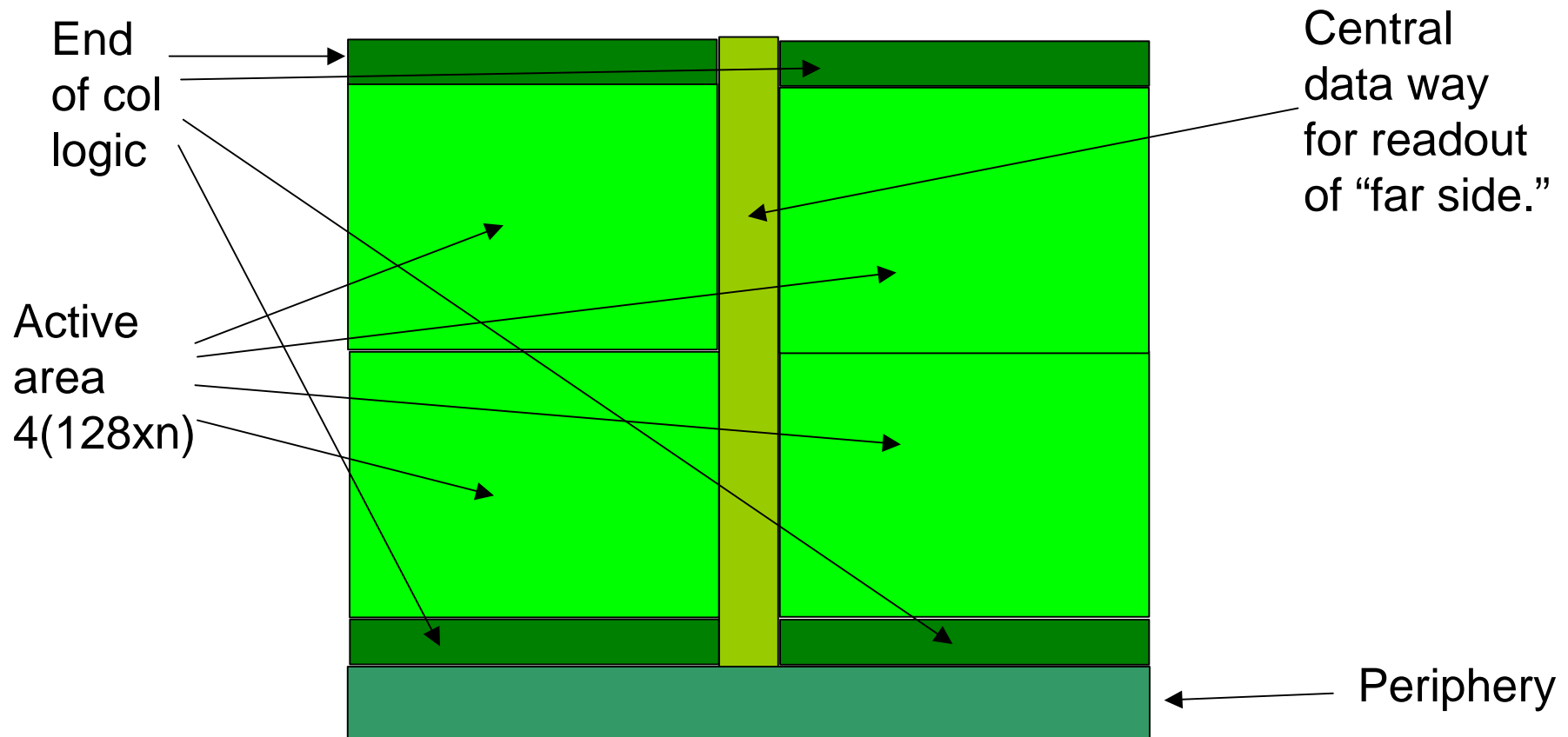
Mixed (256 x n) & (128 x n)

- Can't use only (256xn) because of FPIX2 periphery.
- Larger cantilever required... need to study cooling!
(cooling no problem for solid substrate)



Trying to make $(256 \times n)$ work...

- Lay out 4 “cores” – e.g. $4(128 \times n)$:



Problems with 4(128 x n)

- Still need metal 6 for power distribution (→no TSMC)
 - MIGHT be able to stretch pixels to $\sim 460\mu$ & use extra area for power distribution (Major redesign of core).
- EOC would need redesign to drive central data way
 - MIGHT limit readout speed.
- Much more complex periphery required.
 - Either 4 parallel output paths, or logic to merge output paths is required.
 - Chip would either have to get longer (requiring an even larger cantilever), or much wider, to have room for the extra peripheral logic.

Material Budget

- Assume material is 1/3 sensor, 1/3 readout chips, & 1/3 everything else:
- (256 x n) w/readout on one side is ~7% less material than (128 x n).
 - 12% less in sensors; 9% less in readout chips (assuming no change in periphery size).
- Mixed (128 x n) & (256 x n) w/readout on both sides is ~3% less material than (128 x n).
 - 10% less in sensors; readout chips are the same (assuming no change in periphery size).

How many options shall we pursue?

- 160 x n? (might need more than 18 to fit periphery &/or insure one row of wire bond pads)
- 128 x n? ; mirrored to achieve 256 x n?
 - 128 x n tiles nicely, easiest chip design.
 - Mirrored 256 x n would reduce parts count with respect to 128 x n, but requires larger cantilever & nets only ~3% reduction in material.
- 4(128 x n)?
 - Reduces material by ~7% with respect to 128 x n; also reduces parts count.
 - Requires core redesign (either IBM only design, or pixel longer than 400 μ); speed penalty associated with driving central data way.
- 256 x n with major core redesign?
 - Reduces material by ~7% with respect to 128 x n; also reduces parts count.
 - Major core redesign (either IBM only design, or pixel longer than 400 μ [even longer than 4(128 x n)]).